ECE 486: Control Systems

Lecture 10A: Dominant Pole Approximation
The dominant poles of a higher-order system are the slowest poles (largest time constant).

We can often approximate a higher-order system by a:

1. First-order approximation if the dominant pole is real
2. Second-order approximation if the dominant pole(s) are a complex pair.

The approximation is accurate if the dominant pole(s) are significantly slower than the remaining poles.

(Dominant pole time constant is 5x larger than other poles)
Problem 1

For each system:

• Construct a first-order or second-order approximation from the dominant pole.

• Do you expect the dominant pole approximation to be accurate?

For system A: Roughly sketch the unit step response of the dominant pole approximation. Note the final time, settling time, and overshoot (if underdamped).

\[ G_A(s) = \frac{5000}{(s+2)(s+20)(s^2+20s+500)} \]

\[ G_B(s) = \frac{24}{(s+1)(s+2)^2(s+3)} \]
Solution 1A

- Construct a first-order or second-order approximation from the dominant pole.
- Do you expect the dominant pole approximation to be accurate?
- Roughly sketch the unit step response of the dominant pole approximation. Note the final time, settling time, and overshoot (if underdamped).

\[ G_A(s) = \frac{5000}{(s+2)(s+20)(s^2+20s+500)} \]

\[ s = \begin{bmatrix} -2 & -20 \end{bmatrix} \]

Dominant Pole

\[ p < 1 \rightarrow \text{Pun (Re<1)} \]

\[ -\text{lo} + 20j \]

\[ c = \frac{1}{2} \text{ sec} \]

\[ G_A(0) = \frac{5000}{2 \cdot 20 \cdot 500} = \frac{1}{4} \]

\[ \hat{G} = \frac{b_0}{s+2} \]

\[ \hat{G}(0) = \frac{b_0}{2} \]

\[ b_0 = \frac{2}{4} = \frac{1}{2} \]

Accurate because \(-2\) is much slower than \(-10 \pm 20j\)
Solution 1B

• Construct a first-order or second-order approximation from the dominant pole.
• Do you expect the dominant pole approximation to be accurate?

\[ G_B(s) = \frac{24}{(s+1)(s+2)^2(s+3)} \]

Solutions:
- \( s = -1, -2, -2, -3 \)
- Slowest, dominant

\[ G = \frac{2}{s+1} \]

For approximation:
- \( s = -1, -2, -2, -3 \) are all of similar time scale

\[ G_c(s) = \frac{15}{(s+1)^2(s+10)} \]

Solutions:
- \( s = -1, -1, -10 \)
- Dominant

\[ \hat{G} = \frac{15}{(s+1)^2} \]

\[ \hat{G} = \frac{20}{(s+1)^2} \]

\[ y_p + C_1 e^{-t} + C_2 e^{-3t} \]
ECE 486: Control Systems

Lecture 10B: Integrator Anti-windup
Key Takeaways

This lecture describes impact of actuator saturation and rate limits. These limits:

• Cause slower speed of response and
• Can lead to overshoot and oscillations if the controller does not properly account for the limits.

Anti-windup is one method to reduce the effect of saturation.

• It will prevent overshoot and oscillations.
• However, it does not change the slower speed of response which is a physical limit of the actuators.
Consider the following plant and PI controller:

\[
\dot{y}(t) + 4y(t) = 2u(t) \quad u(t) = 5e(t) + 20 \int_0^t e(\tau) \, d\tau
\]

A) What is the ODE that models the closed-loop from \( r \) to \( y \)?

B) The actuator saturates at \( u \in [-3, +3] \). Do you expect saturation to cause any issues if the reference commands are in the range \( r \in [-1, +1] \)? If yes, then how might you alleviate the issue?

C) Suppose instead that the references are in the range \( r \in [-10, +10] \). Can any controller (not just the PI controller above) achieve good reference tracking with this actuator? If not, then how would you re-select the actuator?
Consider the following plant and PI controller:

\[ \dot{y}(t) + 4y(t) = 2u(t) \quad u(t) = 5e(t) + 20 \int_0^t e(\tau) \, d\tau \]

A) What is the ODE that models the closed-loop from \( r \) to \( y \)?

\[ \begin{align*}
\ddot{y} + 4\dot{y} &= 2\dot{u} = 2 \left[ 5e + 20e \right] \\
\ddot{y} + (4 + 10)\dot{y} + 40y &= 10r + 40r \\
\end{align*} \]

\[ \text{Tr-y(s)} = \frac{10s + 40}{s^2 + 14s + 40} \]
Consider the following plant and PI controller:

\[ \dot{y}(t) + 4y(t) = 2u(t) \quad u(t) = 5e(t) + 20 \int_0^t e(\tau) \, d\tau \]

B) The actuator saturates at \( u \in [-3, +3] \). Do you expect saturation to cause any issues if the reference commands are in the range \( r \in [-1, +1] \)? If yes, then how might you alleviate the issue?

\[
\begin{align*}
  u(t) &= 5e(t) + 20 \int_0^t e(\tau) \, d\tau \\
  u(0) &= 5(r(0) - y(0)) + 20 \int_0^0 e(\tau) \, d\tau \\
  &\text{Clamping / Anti-windup} \\
  &\text{Turn off integration when } |u(t)| > 3 \\
  &\text{(and error makes } |u| \text{ grow more)}
\end{align*}
\]
Solution 2C

Consider the following plant and PI controller:

\[
\dot{y}(t) + 4y(t) = 2u(t) \quad \Rightarrow \quad u(t) = 5e(t) + 20 \int_0^t e(\tau) \, d\tau
\]

C) Suppose instead that the references are in the range \( r \in [-10, +10] \). Can any controller (not just the PI controller above) achieve good reference tracking with this actuator? If not, then how would you re-select the actuator?

\[
\begin{align*}
\text{(In steady-state)} & \\
\frac{1}{2} & = \frac{\bar{y}}{\bar{u}} \quad \Rightarrow \quad \bar{y} = G(s) \bar{u} \\
G(s) &= \frac{2}{s+4} \\
G(0) &= \frac{2}{4} = 0.5 \\
\bar{y} &= G(0) \bar{u} = 0.5 \cdot 3 = 1.5
\end{align*}
\]

Embedded Control
ECE 486: Control Systems

Lecture 10C: Control Law Implementation
Key Takeaways

It is common to implement controllers on a microprocessor. This lecture discusses some of the details associated with this implementation:

• Sample a measurement at specific (discrete) time intervals
• Update the control input \( u \) at each sample time.
• Hold the control input \( u \) constant until the next update.

The update equation is chosen to approximate the properties of the designed (ODE) controller. The update equation can be implemented on a microprocessor with a few lines of code.
Problem 3

Consider the following plant and PI controller:

\[ 2 \dot{y}(t) + 6 y(t) = 8 u(t) \quad u(t) = 2.5 e(t) + 9 \int_0^t e(\tau) \, d\tau \]

A) What sampling time \( \Delta t \) would you recommend for a discrete-time implementation?

B) The value of \( u(t) \) at \( t=\Delta t \) is:

\[ u(\Delta t) = 2.5 e(\Delta t) + 9 \int_0^{\Delta t} e(\tau) \, d\tau \]

Approximate \( u_1 := u(\Delta t) \) in terms \( e_0 := e(0) \) and \( e_1 := e(\Delta t) \).

C) The computations in the discrete-time update are not instantaneous and require some time. How can this be modeled?
Solution 3A

\[ 2 \dot{y}(t) + 6 y(t) = 8 u(t) \quad u(t) = 2.5 e(t) + 9 \int_0^t e(\tau) \, d\tau \]

A) What sampling time \( \Delta t \) would you recommend for a discrete-time implementation?

\[ 2 \dot{y} + 6 y = 8 u = 8 \left[ 2.5 (r-y) + 9 (r-y) \right] \]

\[ 2 \ddot{y} + (6+20) \dot{y} + 72 y = 20 \ddot{r} + 72 \dot{r} \]

\[ T_{r-y}(s) = \frac{20s + 72}{s^2 + 26s + 72} \]

\[ s_{1,2} = -4 \pm \frac{1}{9} \text{ rad/sec} \]

\[ T_{1,2} = \frac{1}{4} \text{ sec}, \frac{1}{9} \text{ sec} \]

\[ \Delta t = \frac{\sqrt{4}}{10} = \frac{1}{10} \text{ sec} \times \]

\[ \Delta t = \frac{1}{10} \approx \frac{1}{90} \text{ sec} = \frac{1}{100} \text{ sec} = 10 \text{ msec} \]
Solution 3B

\[ 2\ddot{y}(t) + 6\dot{y}(t) = 8u(t) \]

\[ u(t) = 2.5e(t) + 9 \int_0^t e(\tau) \, d\tau \]

B) The value of \( u(t) \) at \( t=\Delta t \) is:

\[ u(\Delta t) = 2.5e(\Delta t) + 9 \int_0^{\Delta t} e(\tau) \, d\tau \]

Approximate \( u_1 := u(\Delta t) \) in terms \( e_0 := e(0) \) and \( e_1 := e(\Delta t) \).
The computations in the discrete-time update are not instantaneous and require some time. How can this be modeled?